C212/A592 Lab 3

Intro to Software Systems

Objectives:

- Implementing static method
- Reading values from keyboard (Using Scanner class)
- Using a simple loop
- Casting
- Working with strings
- Junit Testing

Lab instructions

1. Setup for Interactive Textbook Exercises: (**Worth 5 points Bonus**)
   a) Visit the page "*How do I read the textbook, complete Quizzes and complete interactive exercises?*" on Canvas (found within Learning Modules for Week 1-3) o click HERE and familiar yourself with the setup instructions. You should have received an email with the access code. Follow the instructions, and setup your eText access.
   b) Complete atleast 2 Self-Check exercises for Chapter 2 or 4 and show your ScoreCenter to AI/UI. To receive full credit for this exercise, you must have 100% in both of these exercises and you must SHOW them to AI/UI during lab or office hours.
   c) If you run into any problems or can't see your grades in ScoreCenter (within VitalSource), talk to an AI. Be sure to read the instruction carefully.

2. Create a class named Lab3Exercises and Implement the following static methods (we will not use this class to *instantiate objects*) and write **JUnit tests** when asked:

   a) *public static int my1089Puzzle(int myInput)*
      i. Implement a function that receives an integer from user (in main method), and does the following: Discard all but last 3 digits. Reverse the digits, subtract the original (input) from the reversed (discarding any minus sign), reverse the digits of the difference, and add the difference and the reversed difference. Then return the sum (which should be printed in main). For examples:
      Input: 371
      Reversed: 173
      Difference: 198
      Reversed: 891
      Sum: 1089
      For smaller numbers, assume preceding zeros.

      ii. Use the *Scanner* class to get input from the user (in main method) and pass that as a parameter to the method above. Get the returned value and print it along with the user input. Write **JUnit tests**

b) *public static String formatPhoneNumber(String myNumber, boolean option)*

   i. Write the pseudocode for the program

   ii. Implement a static Java method which reads a telephone number into a string variable, computes the formatted number based on user's choice of format and returns it as string. Example:
   If option is *true,* 2342342234 will become (234) 234-2234
   If option is false 2342342234 will become 234-234-2234

   iii. Test your method for both options. Write **JUnit tests**

c) *public static String numbersList()*

   i. Using the *Scanner* class and a while loop, continue to process integer values from a user until they have entered a value other than an *int.* To keep it simple, let's assume that numbers are between 5 and 19. If user enters an integer value outside the given range, prompt the user to enter the correct value again!

   ii. Excluding any duplicates values, Return a string that includes a comma separated list of all user provided integers.

   iii. Test your method (You may use Scanner inside the method)

d) *public static String grade(char)*

   i. Write a method that will process a character value between [A – F]. Return a String value of the range of numerical grades. e.g., input = A returns *"Your grade is between a 93.0 and 100"*

      1. *grade greater than 93.0 –A*
      2. *grade greater than 83.0 –B*
      3. *grade greater than 73.0 –C*
      4. *grade greater than 63.0 –D*
      5. *grade less than 63.0 –F*

   ii. *Test your method for different inputs. Ignore any characters outside A-F.* Write **JUnit tests**

e) *public static String intToHex(int n)*

   i. Implement a method that converts the positive integer value to its hexadecimal representation and returns it as a *String. Hexadecimal number* have 0x preceding the hex digits*. You must print minimum of 2 hex digits for any number.* **Note**: Do not use the java method *Integer.toHexString(N)*
   Example:
   intToHex (1) returns a string 0x01
   intToHex (10) returns a string 0x0A
   intToBinary(16) returns a string 0x10

   ii. Test your program and Write **JUnit tests.** Here are two very good explanations to converting an integer value to binary
   https://www.wikihow.com/Convert-from-Decimal-to-Hexadecimal

3. Answer the following questions as comments below your class file or submit a text file:

a. Give the type and value for each of the following expressions:

i. 7.8 > 78
ii. 12+13+19+21
iii. 2.3+5.7+7.1
iv. 2+4+”24”

b. Without compiling/running the code, give the value printed by each of the following code fragments. Now compile/run and see if your answer was correct. Report the output and whether you were initially correct:

i. The following code is using Newton’s Method to calculate the square root of a number

```
double t = 9.0;
while (Math.abs(t – 9.0/t) > .001) {
          t = (9.0/t + t) / 2.0;
}
System.out.println(t);
```

ii.

```
System.out.printf(“%03d”,7);
```

iii.

```
String s = “Bye Bye Bye”
s = s.replace(‘y’, ‘e’);
System.out.println(s);
```